# RELIABILITY ANALYSIS OF ON-LINE COMPUTER
# SYSTEMS USING COMPUTER - ALGEBRAIC MANIPULATIONS

## BY

## RAHUL CHATTERGY

JANUARY 1976

TECHNICAL REPORT A76-1

82  12  28  082

RELIABILITY ANALYSIS OF ON-LINE COMPUTER

SYSTEMS USING COMPUTER-ALGEBRAIC MANIPULATIONS


by

Rahul Chattergy
University of Hawaii
Honolulu, Hawaii

January 1976

# RELIABILITY ANALYSIS OF ON-LINE COMPUTER

## SYSTEMS USING COMPUTER-ALGEBRAIC MANIPULATIONS

by

Rahul Chattergy
University of Hawaii
Honolulu, Hawaii

## Abstract

This paper discusses the reliability of operation of an on-line computer system described by a semi-Markov process model. Analytical solutions are obtained by using computer-aided algebraic manipulation techniques. This paper demonstrates that the difficulties of obtaining analytical solutions to Markov processes by standard techniques can be considerably reduced by the application of algebraic symbol manipulation languages. To the author's knowledge, the results of the reliability analysis are also new.

A76-1

# I. INTRODUCTION

Consider an on-line computer system as shown in Fig. 1. The users of such a system gain access via the terminals. The computational needs of a user are satisfied by the main processor. Communications among the users, and between any user and the main processor, are supervised by the front-end processor. The main purpose of a front-end processor is to relieve the main processor of the communications-management duties. In such a system, either the front-end processor or the main processor or both may fail at any time. We assume that if the front-end processor fails, then the communications-management job is taken over by the main processor. This causes an increase in response-time as observed by the users and constitutes a degradation of the system-performance. If the main processor fails, then evidently the users have a limited computational support from the front-end processor but they may communicate with each other via the front-end processor. This situation constitutes a more serious degradation of the system-performance than the one mentioned above. Finally, if both processors fail, the system is shut down completely for repairs. In such a situation, the main processor has priority on repair service over the front-end processor. Note that the system is not a simple special case of the repairman problem [2] or a series connection of two unreliable subsystems [3].

This system can be characterized by four states which are listed in Table 1. Transitions between pairs of states occur at randomly distributed instants of time. Therefore, it is possible to describe the system by a semi-Markov process. The reliability of operation of such a system can be measured by
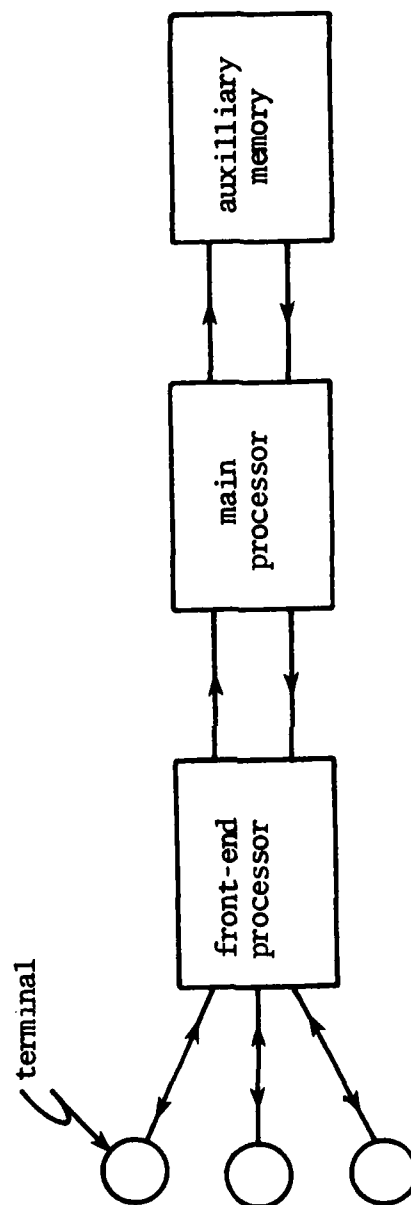
Figure 1

DIAGRAM OF SYSTEM

various probabilistic measures related to the semi-Markov process. In the next section we describe the semi-Markov process in more detail.

## II. SEMI-MARKOV PROCESS MODEL

A semi-Markov process is described by a set of states and probabilistic descriptions of the temporal transitions between pairs of states [1]. The on-line computer system has four states, each state uniquely labelled by an integer from the set [1,...,4] as shown in Table 1. Allowable transitions between pairs of states are shown in Fig. 2. Let $t_i$ denote the time spent by the process in state i before a transition to some other state occurs. We define the waiting-time distribution in state i as

$$W_i(t) = P[t_i \leq t] ,$$

and the corresponding density function and mean by $w_i(t)$ and $\bar{w}_i$, respectively.

Let $[0,F_1]$ and $[0,F_2]$ respectively denote the failure-free operation intervals of the main and front-end processors. Following arguments in [2] and [3], we assume the failure-time distributions to be

$$P[F_1 \leq t] = 1 - \exp(-L_1 t) ,$$

and
$$P[F_2 \leq t] = 1 - \exp(-L_2 t) ,$$

for $t \geq 0$ and zeros otherwise. Let $[0,R_1]$ and $[0,R_2]$ denote the repair-time intervals of the main and front-end processors, respectively. We assume the repair-time distributions to be

$$P[R_1 \leq t] = 1 - \exp(-G_1 t) ,$$

and
$$P[R_2 \leq t] = 1 - \exp(-G_2 t) ,$$
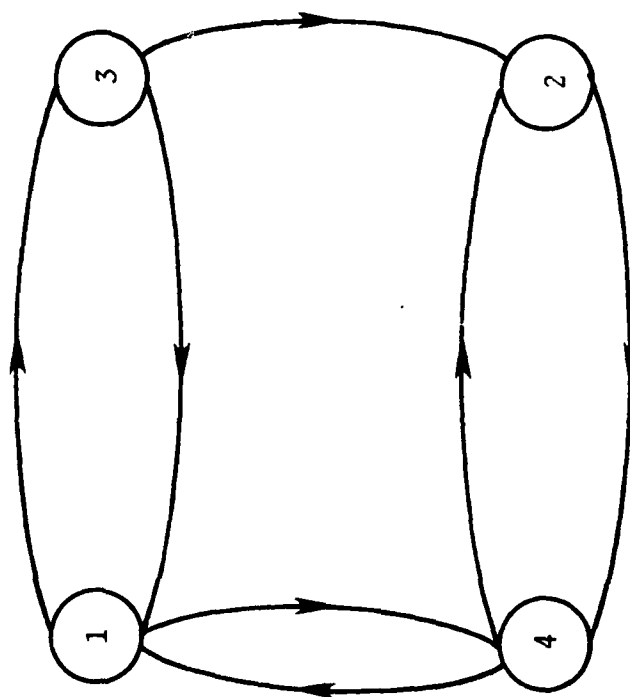
Figure 2

STATE TRANSITION DIAGRAM

for $t \geq 0$ and zeros otherwise. Note that since the main processor has priority of repair service, only the exponential distribution for $P[R_2 \leq t]$ will result in a Markov process. The waiting-time distribution for each state can now be evaluated in a straightforward manner. As an example, let us evaluate $W_3(t)$. We define the following events:

$E_1$: Repair-time of the main processor $> t$ ,

$E_2$: Failure-time of the front-end processor $> t$ .

Then

$$1 - W_3(t) = P[E_1 \text{ and } E_2] ,$$
$$= P[E_1] \, P[E_2] ,$$
$$= \exp(-G_1 t) \, \exp(-L_2 t) ,$$
$$= \exp(-(G_1 + L_2) t) ;$$

and hence

$$W_3(t) = 1 - \exp(-(G_1 + L_2) t) .$$

The corresponding density function is

$$w_3(t) = (G_1 + L_2) \exp(-(G_1 + L_2) t) ;$$

and the mean is

$$\overline{w}_3 = 1/(G_1 + L_2) .$$

A list of the four waiting-time distributions, their density functions and means are given in Table 2.

Let $p_{ij}(t)$ denote the conditional probability density function of a transition to state j in $[t,t+\Delta]$ given that the process entered state i at time zero and the next transition from state i occurs in $[t,t+\Delta]$ for sufficiently small $\Delta>0$. Then the probability density function of a transition from state i to state j after waiting t units of time in state i is given by

$$c_{ij}(t) = p_{ij}(t)w_i(t) \ .$$

The core matrix of a semi-Markov process is defined as $C(t)=[c_{ij}(t)]$, and it provides a complete probabilistic description of the process [1].

Let us consider a sample calculation for the conditional transition probability density function from state 3 to state 2. We define the following events:

$E_3$: Front-end processor fails in $[t,t+\Delta]$ ,

$E_4$: Main processor not repaired in $[0,t]$ ,

$E_5$: Transition occurs out of state 3 in $[t,t+\Delta]$ .

Then

$$p_{32}(t) = P[(E_3 \text{ and } E_4)/E_5] \ ,$$

$$= P[(E_3 \text{ and } E_4) \text{ and } E_5]/P[E_5] \ ,$$

$$= P[E_3 \text{ and } E_4]/P[E_5] \ ,$$

$$= \frac{(\Delta L_2 \exp(-L_2 t))\exp(-G_1 t)}{\Delta(G_1+L_2)\exp(-(G_1+L_2)t)} \ ,$$

$$= L_2/(G_1+L_2) \ .$$

The core matrix of this process is shown in Table 3. The single-step transition probability matrix of the imbedded Markov chain is given by $P = \int_0^\infty P(t)dt$. The P matrix for this process is shown in Table 4.

Let $e_{ij}(t)\Delta$ denote the probability that the process will enter state j in $[t, t+\Delta]$ given that it was in state i at time zero and let $E(t) = [e_{ij}(t)]$. Then

$$E(s) = [I - C(s)]^{-1}$$

where I is the identity matrix and $E(s)$ and $C(s)$ are Laplace transforms of $E(t)$ and $C(t)$ respectively (see [1]). Define

$$E = \lim_{s \to 0}[sE(s)] .$$

For a monodesmic process such as the one described here, the rows of E are identical [1]. Let $e_j$ denote the $j^{th}$ element of any row of E. Then the limiting interval transition probability for state j, denoted by $h_j$, is given by

$$h_j = e_j \overline{w}_j .$$

Suppose the process has been operating unobserved for a long period of time. Then $h_j$ is the probability of the event that the process will be in state j when observed next. The state occupancy statistics can also be obtained from $E(s)$. The details of this and mean first-passage time computations can be found in [1].

The next section shows how analytic expressions can be obtained for $h_j$, state occupancy statistics, and mean first-passage times by using computer-aided algebraic manipulation techniques. This material was generated interactively, by using the symbol manipulation language called MACSYMA available

## Table 1

### STATE DESCRIPTION TABLE

| STATE | LABEL |
|---|---|
| Both processors in operation | 1 |
| Both processors down | 2 |
| Only main processor down | 3 |
| Only front-end processor down | 4 |

## Table 2

### WAITING-TIME DISTRIBUTIONS

| STATE | DISTRIBUTION | DENSITY | MEAN |
|---|---|---|---|
| 1 | $1 - \exp(-(L_1+L_2)t)$ | $(L_1+L_2)\exp(-(L_1+L_2)t)$ | $1/(L_1+L_2)$ |
| 2 | $1 - \exp(-G_1 t)$ | $G_1\exp(-G_1 t)$ | $1/G_1$ |
| 3 | $1 - \exp(-(G_1+L_2)t)$ | $(G_1+L_2)\exp(-(G_1+L_2)t)$ | $1/(G_1+L_2)$ |
| 4 | $1 - \exp(-(L_1+G_2)t)$ | $(L_1+G_2)\exp(-(L_1+G_2)t)$ | $1/(L_1+G_2)$ |

## Table 3

### CORE MATRIX

| | | | |
|---|---|---|---|
| 0 | 0 | $L_1\exp(-(L_1+L_2)t)$ | $L_2\exp(-(L_1+L_2)t)$ |
| 0 | 0 | 0 | $G_1\exp(-G_1 t)$ |
| $G_1\exp(-(G_1+L_2)t)$ | $L_2\exp(-(G_1+L_2)t)$ | 0 | 0 |
| $G_2\exp(-(L_1+G_2)t)$ | $L_1\exp(-(L_1+G_2)t)$ | 0 | 0 |

Table 4

TRANSITION PROBABILITY MATRIX OF THE IMBEDDED MARKOV CHAIN

| $0$ | $0$ | $L_1/(L_1+L_2)$ | $L_2/(L_1+L_2)$ |
|---|---|---|---|
| $0$ | $0$ | $0$ | $1$ |
| $G_1/(G_1+L_2)$ | $L_2/(G_1+L_2)$ | $0$ | $0$ |
| $G_2/(L_1+G_2)$ | $L_1/(L_1+G_2)$ | $0$ | $0$ |

at the MIT mathematics laboratory, accessed via the ARPA computer-communications network.

### III. INTERACTIVE ANALYSIS USING MACSYMA

Comments imbedded between /* and */ are not part of the interactive session, but have been added to explain the process.

```
/* TIME:TRUE WILL PRINT CPU TIME USED IN EACH STEP:  */
/* A,B,C ARE CONSTANTS USED TO SIMPLIFY DATA INPUT:  */
```

(C1)               TIME:TRUE $

TIME= 8 MSEC.

(C2)               A:L1/(L1+L2) $

TIME= 35 MSEC.

(C3)               B:G1/(G1+L2) $

TIME= 28 MSEC.

(C4)               C:G2/(L1+G2) $

TIME= 25 MSEC.

```
/* WAIT:ROW-VECTOR OF MEAN WAITING TIMES. ELEMENT I     */
/* IS THE MEAN WAITING TIME AT STATE I:                 */
```

(C5)               WAIT:MATRIX([A/L1,1/G1,B/G1,C/G2]);

TIME= 60 MSEC.

$$\begin{bmatrix} \dfrac{1}{L2 + L1} & \dfrac{1}{G1} & \dfrac{1}{L2 + G1} & \dfrac{1}{L1 + G2} \end{bmatrix}$$

(D5)

```
/* TCORE IS THE CORE MATRIX OF THE SEMI-MARKOV PROCESS    */
/* TCORE IS PRINTED ROW BY ROW, EACH ROW ENCLOSED IN [ ]. */
/* %E DENOTES THE EXPONENTIAL "e".  ^ DENOTES EXPONENTIATION:  */
```

(C6)

TCORE:MATRIX

([0,0,L1*%E^((-L1/A)*T),L2*%E^((-L1/A)*T)],
 [0,0,0,G1*%E^(-G1*T)],
 [G1*%E^((-G1/B)*T),L2*%E^((-G1/B)*T),0,0],
 [G2*%E^((-G2/C)*T),L1*%E^((-G2/C)*T),0,0]);

TIME= 468 MSEC.

$$(D6) \quad MATRIX([0, 0, L1 \, \%E^{(-L2-L1)T}, L2 \, \%E^{(-L2-L1)T}],$$

$$[0, 0, 0, G1 \, \%E^{-G1T}], [G1 \, \%E^{(-L2-G1)T}, L2 \, \%E^{(-L2-G1)T}, 0,$$

$$0], [G2 \, \%E^{(-L1-G2)T}, L1 \, \%E^{(-L1-G2)T}, 0, 0])$$

/* THE NESTED DO LOOPS COMPUTE THE LAPLACE TRANSFORM OF    */
/* TCORE AND STORE IT IN A TEMPORARY ARRAY TEMPO:          */
(C7)

```
        FOR I:1 THRU 4 DO
            FOR J:1 THRU 4 DO
                TEMPO[I,J]:LAPLACE(TCORE[I,J],T,S);
```

LAPLAC FASL DSK MACSYM BEING LOADED
LOADING DONE
TIME= 1195 MSEC.
(D7)                         DONE

/* GENMATRIX OPERATION IS USED TO CONVERT ARRAY TEMPO      */

/* INTO MATRIX SCORE=LAPLACETRANSFORM( TCORE)              */
/* SCORE MATRIX SHOWN BELOW :                              */

(C8)        SCORE:GENMATRIX(TEMPO,4,4);

GENMAT FASL DSK MAXOUT BEING LOADED
LOADING DONE
TIME= 98 MSEC.

$$
\begin{bmatrix}
0 & 0 & \dfrac{L1}{S+L2+L1} & \dfrac{L2}{S+L2+L1} \\
0 & 0 & 0 & \dfrac{G1}{S+G1} \\
\dfrac{G1}{S+L2+G1} & \dfrac{L2}{S+L2+G1} & 0 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
\dfrac{G2}{S + L1 + G2} & \dfrac{L1}{S + L1 + G2} & 0 & 0
\end{bmatrix}
$$

(C9)                 KILL(TEMPO);

TIME= 8 MSEC.

(D9)            DONE

/* SENTRY=INVERSE OF ( IDENTITY - SCORE ).   ^^ DENOTES NONCOMMUTATIVE   */
/* EXPONENTIATION.   INVERSE OF MATRIX = MATRIX ^^-1:                     */
(C10)

SENTRY:(IDENT(4)-SCORE)^^-1 $

TIME= 82298 MSEC.

/* TEMPO= S * FIRST ROW OF SENTRY.  RATSIMP SIMPLIFIES EXPRESSIONS:    */

(C11)          TEMPO:RATSIMP(S*ROW(SENTRY,1)) $

TIME= 167759 MSEC.

/* LMT= LIMIT OF TEMPO AS S-->0:                        */

(C12)          LMT:LIMIT(TEMPO,S,0) $

LIMIT FASL DSK MACSYM BEING LOADED
LOADING DONE
TIME= 2873 MSEC.

/* LMTDST= LIMITING INTERVAL TRANSITION PROBABILITIES      */
/* OF THE SEMI-MARKOV PROCESS:                            */

(C13)          LMTDST:RATSIMP(LMT*WAIT) $

TIME= 28120 MSEC.

/* CHECK ON LMTDST FOR L1=0 AND L2=0.  EVALUATE LMTDST FOR L2=0:   */

(C14)          LMTDST,L2=0 $

TIME= 683 MSEC.

(C15)          RATSIMP(D14);

TIME= 997 MSEC.

(D15)
$$\begin{bmatrix} \dfrac{G1}{L1+G1} & 0 & \dfrac{L1}{L1+G1} & 0 \end{bmatrix}$$

```
/* EVALUATE LMTDST FOR L1=0:                              */
(C16)              LMTDST,L1=0 $

TIME= 646 MSEC.

(C17)              RATSIMP(D16);

TIME= 4652 MSEC.
```

(D17)
$$\begin{bmatrix} \dfrac{G2}{L2+G2} & 0 & 0 & \dfrac{L2}{L2+G2} \end{bmatrix}$$

```
/* LIMITING INTERVAL TRANSITION PROBABILITIES IN    */
/* GENERAL FORM SHOWN BELOW :                       */

(C18)            LMTDST;

TIME= 2 MSEC.
```

(D18) $\text{MATRIX}([(G1\ G2\ L2 + G1^2\ G2)/((L1 + G1)\ L2^2$

$+ (L1^2 + (G2 + 2\ G1)\ L1 + G1\ G2 + G1^2)\ L2 + G1\ G2\ L1 + G1^2\ G2),$

$(L1\ L2^2 + (L1^2 + (G2 + G1)\ L1)\ L2)/((L1 + G1)\ L2^2$

$+ (L1^2 + (G2 + 2\ G1)\ L1 + G1\ G2 + G1^2)\ L2 + G1\ G2\ L1 + G1^2\ G2),$

$G1\ G2\ L1/((L1 + G1)\ L2^2 + (L1^2 + (G2 + 2\ G1)\ L1 + G1\ G2 + G1^2)\ L2$

$+ G1\ G2\ L1 + G1^2\ G2),\ (G1\ L2^2 + (G1\ L1 + G1^2)\ L2)$

$/((L1 + G1)\ L2^2 + (L1^2 + (G2 + 2\ G1)\ L1 + G1\ G2 + G1^2)\ L2 + G1\ G2\ L1$

$+ G1^2\ G2)])$

```
/* COMPUTE STATE OCCUPANCY STATISTICS. TOCUP[1,2]   */
/* = AVRG. NO. OF TIMES STATE 2 IS VISITED IN [0,T] */
/* STARTING IN STATE 1 AT TIME ZERO. ILT = INVERSE  */
/* LAPLACE TRANSFORM :                              */
```

(C19)        SOCUP[1,2] = RATSIMP(SENTRY[1,2]/S) $

TIME= 5426 MSEC.

(C20)        TOCUP[1,2] = ILT(SOCUP[1,2],S,T) $
/* INFORMATION REQUESTED BY THE ILT ROUTINE:           */
IS $L1^2 + (2G2 + 2G1)L1 + G2^2 - 2G1G2 + G1^2$
                    POSITIVE, NEGATIVE, OR ZERO?
/* ANSWER ENTERED FROM THE TERMINAL:           */
POSITIVE;

TIME= 162735 MSEC.

(C21)        TOCUP[1,2],L1=1,L2=3,G1=10,G2=30;

TIME= 5377 MSEC.

(D21)
$$\%E^{-21T}\left(\frac{114998\ \text{SINH}(\text{SQRT}(174)\ T)}{134657\ \text{SQRT}(174)} + \frac{9515\ \text{COSH}(\text{SQRT}(174)\ T)}{134657}\right)$$
$$+ \frac{65\ T}{178} - \frac{3\ \%E^{-6T}}{68} - \frac{841}{31684}$$

/* NOTE THAT TOCUP[1,2] HAS A LINEAR TERM IN T     */
/* FOR LARGE T THIS TERM WILL BE DOMINANT :        */

(C22)        PART(TOCUP[1,2],2);

TIME= 3 MSEC.

(D22)        $(G1\ L1\ L2^2 + (G1\ L1^2 + (G1\ G2 + G1^2)\ L1)\ L2)\ T$

$/((L1 + G1)\ L2^2 + (L1^2 + (G2 + 2G1)\ L1 + G1\ G2 + G1^2)\ L2$

$+ G1\ G2\ L1 + G1^2\ G2)$

## IV.  DISCUSSION OF THE RESULTS

Two important parameters for estimating the reliability of operation of
this system are $h_1$ and $h_2$, respectively the probabilities of being fully

operational and completely shutdown in the steady state. Let $D=(L_1+G_1)(L_2^2+G_1G_2)$ $+(L_1^2+(G_2+2G_1)L_1+G_1(G_1+G_2))L_2$. Then

$$h_1 = G_1G_2(G_1+L_2)/D, \tag{1}$$

and
$$h_2 = L_1L_2(L_1+L_2+G_1+G_2)/D. \tag{2}$$

Equations (1) and (2) show the dependence of $h_1$ and $h_2$ on all the parameters of the system. Let us investigate the effects of $L_1$ and $L_2$ on $h_1$. Note that $L=0$ means the corresponding processor never fails. As $L$ goes to infinity, the corresponding processor fails with probability one in any time interval of positive length. If $L_1=0$, then

$$h_1 = G_2/(L_2+G_2), \tag{3}$$

$$= 1/(1+(L_2/G_2),$$

$$= 1/(1+(\text{Average repair-time/Average failure-time}) ;$$

and if $L_2=0$, then

$$h_1 = G_1/(L_1+G_1), \tag{4}$$

$$= 1/(1+(L_1/G_1)) ,$$

$$= 1/(1+(\text{Average repair-time/Average failure-time}).$$

For either $L_1=0$ or $L_2=0$, the process reduces to a simple two-state semi-Markov process and $h_1$ can be easily computed to verify the above results. These results were used as checks during the interactive session with MACSYMA. As $L_2$ increases without bound, $D=G_1L_2^2$ (approximate) and

$$h_1 = G_2/L_2 \text{ (approximate)}, \tag{5}$$

$$= (\text{Average failure-time/Average repair-time}) \text{ for front-end processor;}$$

$$h_2 = L_1/G_1 \text{ (approximate)}, \tag{6}$$

$$= \text{(Average reapir-time/Average failure-time) for main processor.}$$

As $L_1$ increases without bound, $D=L_1^2 L_2$ (approximate) and

$$h_1 = G_1 G_2 (G_2 + L_2) L_1^2 L_2 \text{ (approximate)}, \tag{7}$$

$$h_2 = 1 \text{ (approximate)}. \tag{8}$$

Let us now consider the effects of $G_1$ and $G_2$ on $h_1$. Note that G=0 means that the corresponding processor is never repaired, and as G goes to infinity, the corresponding processor is repaired with probability one in any time interval of positive length. As $G_1$ increases without bound, $D=G_1^2(G_2+L_2)$ (approximate) and

$$h_1 = G_2/(G_2 + L_2) \text{ (approximate)}. \tag{9}$$

Note that equations (3) and (9) give identical values of $h_1$ since no failure of main processor is equivalent to failures followed by instantaneous repairs. As $G_2$ increases without bound, $D=G_1 G_2(G_1 + L_1 + L_2) + G_2 L_1 L_2$ and

$$h_1 = G_1/(L_1 + G_1) \text{ (approximate)}. \tag{10}$$

Equations (4) and (10) give identical values of $h_1$ for reasons explained above.

Next let us consider a state-occupancy statistic of relevance to the reliability of this system. Let $N_{12}(T)$ denote the average number of times the process visits state 2 in the time interval $[0,T]$, starting in state 1 at time zero. Then for large values of T,

$$N_{12}(T) = G_1 h_2 T \text{ (approximate)}. \tag{11}$$

Equation (11) is derived by observing the results obtained interactively with MACSYMA. The exact expression of $N_{12}(T)$ contain exponentially decaying terms of T and the linear term, shown in equation (11). It is interesting to consider the effect of $G_1$ on $N_{12}(T)$. As $G_1$ increases without bound, $D=G_1^2(G_2+L_2)$ (approximate), $h_2=L_1L_2/(G_1(G_2+L_2))$ (approximate) and consequently $G_1h_2=L_1L_2/(G_2+L_2)$. Thus

$$N_{12}(T) = L_1L_2T/(G_2+L_2) \text{ (approximate).} \tag{12}$$

We observe that $L_1L_2/(G_2+L_2)$ is a lower bound on the slope of $N_{12}(T)$ as $G_1$ goes to infinity. Note that even for very large values of $G_1$, the average number of times a complete system-breakdown occurs in [0,T] cannot be made less than that given by equation (12).

Next, let us consider the average first-passage times between pairs of states. Let $T_{ij}$ denote the average first-passage time from state i to state j. Then we define $T_{21}$ as the recovery-time of the system from complete break-down in state 2 to fully operational in state 1. The unknown quantities $T_{ij}$ satisfy a set of linear algebraic equations which can be easily solved by MACSYMA. Solving these equations we have

$$T_{21} = (L_1+G_2)(1/G_1 + 1/(L_1+G_2))/G_2 \ .$$

As $G_1$ increases without bound, recovery-time becomes

$$T_{21} = 1/G_2 \ , \tag{14}$$

Recovery-time = Average repair-time for front-end processor.

As $G_2$ increases without bound, recovery-time becomes

$$T_{21} = 1/G_1,$$

Recovery-time = Average repair-time for main processor.

These two values of $T_{21}$ are intuitively obvious. If the main processor does not fail too soon, i.e., $L_1$ is close to zero, then

$$T_{21} = (1/G_1) + (1/G_2), \tag{15}$$

i.e., Recovery-time = Average repair-time for main processor

+ Average repair-time for front-end processor.

There are other reliability-related expressions which can be studied in a similar manner. We believe that the above results illustrate the usefulness of computer-aided algebraic manipulation techniques for analyzing semi-Markov processes.


## V. CONCLUSION


The main advantage of using computer-aided algebraic manipulation stems from the fact that the core matrix can be a function of system parameters. Closed-form analytical solutions can be obtained as functions of these system parameters. Numerical analysis will either require repetition of the same computations for different parameter values or some sort of sensitivity analysis over a limited range of parameter values. Also there is no need to invent

ingenious *ad-hoc* methods for individual problems.  Using computer algebra, the same standard approach can be used to analyze all such processes.

The size of the system, as measured by the number of states, that can be handled in this manner, obviously depends on the speed and memory-size of the computer being used.  It has been observed that the CPU time required to invert a matrix to compute E(s) is usually much less than the time required for simplifying some results using RATSIMP or the time required for computing the inverse Laplace transform to obtain $N_{ij}(T)$.  Normally only a few expressions of interest need to be computed for any system.  Hence, as long as E(s) can be computed in a reasonable period of time, the corresponding system can be analyzed by this approach.

## REFERENCES

[1] HOWARD, R.A.: *Dynamic Probabilistic Systems, Vol. 2,* John Wiley, 1971.

[2] BARLOW, R.E.; PROSCHAN, F.; and HUNTER, L.C.: *Mathematical Theory of Reliability,* John Wiley, 1965.

[3] GNEDENKO, B. V.; BELYAYEV, Yu.k.; and SOLOVYEV, A.D.: *Mathematical Methods of Reliability Theory,* Academic Press, 1969.

## ACKNOWLEDGEMENTS